

## AMENDMENTS TO THE CLAIMS

### Claims Pending:

- At time of the Action: **Claims 1-3, 5-11, 13, 14, and 16-38**
- Amended Claims: **Claims 1, 8, 17, 27, and 32**
- After this Response: **Claims 1-3, 5-11, 13, 14, and 16-38**

1.     **(Currently Amended)** A method, comprising:

receiving a request to add a new filter to a filter table stored in an inverse query engine cache;

adding the new filter to the filter table, wherein the new filter comprises a condition field, a data field, an expiration field, a filter weight field, and a permanent flag field, the permanent flag field being a Boolean field indicating that the new filter is not to be removed from the filter table during an expire cache operation or a trim cache operation;

assigning a weight value in the filter weight field to the new filter based on an estimate of a size of the new filter, wherein the weight value denotes the relative size of the new filter in relation to other filters stored in the filter table of the inverse query engine;

determining the filter table of a bounded size;

maintaining the inverse query engine cache at or below a maximum cache size, wherein the size of the inverse query engine cache may be indicated by size of the filter table, estimate of size of the filter table, or by cache usage;

wherein the inverse query engine cache comprises a control module, a cache, an add filter module, a remove filter module, a matcher, a maintainer, an expire module, a

trim module, a cache weight module, a cache weight, an optimal weight, a maximum weight, a filter table, a most recently used list, and an expiration list;

wherein the expiration list comprises a filter identifier including an expiration value in the expiration field;

removing a filter based on an expiration time;

checking the expiration value in the expiration field of a filter against a current time, in response to the expiration value is earlier than the current time, remove the filter;

checking the specified period of time, removing a filter that has been stored in the filter table longer than a specified period of time;

trimming the filter table upon the occurrence of the filter table reaching the maximum weight, by determining the cache weight and identifying filters to be removed and removing filters from the filter table to obtain the optimal weight; and

wherein determining when a permanent flag in a filter is set, leave the filter in the filter table;

wherein the inverse query engine cache is used exclusively by an inverse query engine to store filters associated therewith.

2. (Original) The method as recited in claim 1, further comprising maintaining the size of the inverse query engine cache between an optimal cache size and the maximum cache size.

3. (Previously Presented) The method as recited in claim 1, wherein the maintaining further comprises:

determining if the addition of the new filter to the filter table increases the cache size above the maximum cache size; and

removing one or more filters from the filter table if the addition of the new filter causes the cache size to exceed the maximum cache size.

4. (Cancelled)

5. (Original) The method as recited in claim 1, wherein the maintaining further comprises:

identifying a weight associated with the new filter;

adding the weight associated with the new filter to a cache weight that is the sum of filter weights of filters in the filter table, each filter having a filter weight; and

comparing the cache weight to the maximum cache size.

6. (Original) The method as recited in claim 1, further comprising identifying one or more expired filters in the filter table; and wherein the maintaining the inverse query engine cache further comprises removing at least one of the identified expired filters.

7. (Original) The method as recited in claim 1, further comprising a least recently used filter in the filter table; and wherein the maintaining the inverse query engine cache further comprises removing the least recently used filter from the filter table when a size of the inverse query engine cache reaches the maximum cache size.

8. **(Currently Amended)** A system, comprising:

- a memory;
- a processor coupled to the memory;
- an inverse query engine configured to test an input against a collection of filters;
- a cache associated with the inverse query engine, wherein the inverse query engine cache comprises a control module, an add filter module, a remove filter module, a matcher, a maintainer, an expire module, a trim module, a cache weight module, a cache weight, an optimal weight, a maximum weight, a filter table, a most recently used list, and an expiration list;
- the inverse query engine is maintained within a bounded size by removing expired filters from the cache and trimming the cache to an optimal size when the cache reaches a maximum size;
- the filter table stored in the cache and containing multiple filters, wherein the filter comprises a condition field, a data field, an expiration time field, a filter weight field, and a permanent flag field; and
- a maintainer configured to maintain a size of the filter table within definite cache bounds, wherein the trim module of the maintainer, upon the occurrence of the filter table reaching the maximum weight, determines the cache weight and identifies filters to be removed and removes filters from the filter table to obtain the optimal weight automatically, if the maximum weight is exceeded;
- wherein the size of the filter table may be indicated by size of the collection of filters or by weight of the collection of filters, wherein the inverse query engine assigns a

weight value in the filter weight field to the filter based on an estimate of the size of the filter, wherein the weight value denotes the relative size of the filter in relation to other filters of the collection of filters stored in the inverse query engine.

9. (Original) The system as recited in claim 8, further comprising an expiration module configured to remove expired filters from the filter table.

10. (Original) The system as recited in claim 9, wherein the maintainer is further configured to invoke the expiration module when a new filter is added to the filter table.

11. (Original) The system as recited in claim 8, further comprising a trim module configured to remove one or more filters from the filter table when the cache reaches a maximum cache size.

12. (Cancelled)

13. (Original) The system as recited in claim 11, wherein the trim module is further configured to determine if a permanent flag in a filter is set and, if the permanent flag is set, to leave the filter in the filter table.

14. (Original) The system as recited in claim 11, wherein:  
a filter weight is associated with each filter in the filter table;

the cache size further comprises a cache weight that is a sum of all filter weights in the filter table;

the maximum cache size further comprises a maximum cache weight; and

wherein the trim module is further configured to deduct a filter weight from the cache weight when a filter associated with the filter weight is removed from the filter table.

15. (Cancelled)

16. (Original) The system as recited in claim 11, further comprising:

a most recently used list configured to track usage of filter table filters according to when filters are used, and indicating that a filter has been most recently used when the filter is the latest filter to match an input; and

wherein the trim module is further configured to remove the one or more filters from the filter table based on which filters have been least recently used.

17. (Previously Presented) One or more computer-readable storage media storing computer-executable instructions that, when executed on a computer, perform the following acts :

receiving a request to add a new query to an inverse query engine cache that stores multiple queries;

assigning a size value to the new query based on cache usage, size of the new query, or estimate of size of the new query, wherein the size value denotes the relative size of the new query in relation to other queries stored in the inverse query engine cache;

defining conditions and processing input that satisfies the conditions;

deriving a cache size that is a sum of query sizes of the queries stored in the inverse query engine;

determining if the cache size is at greater than or equal to a maximum cache size;

removing one or more queries from the inverse query engine cache if the cache size is greater than or equal to the maximum cache size, by identifying queries to be removed and removing the one or more queries to obtain an optimal size;

deducting the query size of each query removed from the cache size;

adding the new query to the inverse query engine cache; and

adding a new query size to the cache size, the new query size identifying a size of the new query added to the inverse query engine cache;

wherein the inverse query engine cache comprises a control module, a cache, an add filter module, a remove filter module, a matcher, a maintainer, an expire module, a trim module, a cache weight module, the cache size, the optimal size, the maximum cache size, a filter table, a most recently used list, and an expiration list.

18. (Previously Presented) The one or more computer-readable storage media as recited in claim 17, wherein removing further comprises removing a query from the inverse query engine that has been used less recently than other queries stored in the inverse query engine cache.

19. (Previously Presented) The one or more computer-readable storage media as recited in claim 17, wherein a query size is an estimate of the actual size of the query.

20. (Previously Presented) The one or more computer-readable storage media as recited in claim 17, wherein adding the new query size to the cache size is performed before determining if the cache size is greater than or equal to the maximum cache size.

21. (Previously Presented) The one or more computer-readable storage media as recited in claim 17, further comprising instructions to perform the following additional acts :

determining if any queries stored in the inverse query engine cache have expired;  
and  
removing one or more expired queries from the inverse query engine cache.

22. (Previously Presented) The one or more computer-readable storage media as recited in claim 17, wherein the new query size is received with the new query.

23. (Previously Presented) The one or more computer-readable storage media as recited in claim 17, further comprising instructions to perform the additional act of determining the new query size.



24. (Previously Presented) The one or more computer-readable storage media as recited in claim 23, wherein the determining the new query size further comprises estimating the new query size.

25. (Previously Presented) The one or more computer-readable storage media as recited in claim 17, further comprising the additional act of:

determining if a query in the inverse query engine cache includes an indication that the query is a permanent query; and

wherein the query is not removed from the inverse query engine cache if the query is a permanent query.

26. (Previously Presented) The one or more computer-readable storage media as recited in claim 17, wherein a query size is represented as a weight value that denotes the relative size of the query associated therewith with regard to other queries stored in the inverse query engine cache.

27. **(Currently Amended)** A method of maintaining an inverse query engine cache, comprising:

assigning a weight value in a filter weight field to a filter based on cache usage, size of the filter, or estimate of size of the filter, wherein the weight value denotes the relative size of the filter in relation to other filters stored in a filter table of the inverse query engine;

determining when inverse query engine cache usage is approaching a cache usage capacity, wherein the cache usage capacity is determined by a weight of a filter table comprising a condition field, a data field, an expiration time field, a filter weight field, and a permanent flag field; ~~and~~

removing one or more filters from the inverse query engine cache upon the occurrence of the filter table reaching a maximum weight, by determining the cache weight and identifying filters to be removed and removing filters from the filter table to obtain an optimal weight, and

wherein removing one or more filters comprises at least one of expiring or trimming the cache;

checking the expiration field of a filter against a current time, if an expiration value in the expiration field is earlier than the current time, then remove the filter;

wherein an inverse query engine cache comprises a control module, a cache, an add filter module, a remove filter module, a matcher, a maintainer, an expire module, a trim module, a cache weight module, the cache weight, the optimal weight, the maximum weight, the filter table, a most recently used list, and an expiration list.

28. (Previously Presented) The method as recited in claim 27, wherein removing further comprises removing at least one expired filter from the inverse query engine cache.

29. (Original) The method as recited in claim 28, wherein an expired filter is a filter having an expiration time associated therewith that precedes a current time.

30. (Original) The method as recited in claim 28, wherein an expired filter is a filter that has been stored in the inverse query engine cache for a predefined period of time.

31. (Original) The method as recited in claim 27, wherein [[the]] removing [[step]] further comprises removing a least recently used filter from the inverse query engine cache.

32. **(Currently Amended)** An inverse query engine having an integrated cache, the inverse query engine configured to assign a weight value to a filter of the integrated cache based on an estimate of the size of the filter, wherein the weight value denotes the relative size of the filter in relation to other filters of the integrated cache [[.]]:

the inverse query engine is maintained within a bounded size by removing expired filters from the cache and trimming the cache to an optimal size when the cache reaches a maximum size;

wherein the cache is maintained within predefined limits by removing at least one filter from a group of filters stored in the cache that has been used less recently than other filters in the group of filters.

33. (Original) The inverse query engine as recited in claim 32, wherein the cache is bound to a finite size.

34. (Original) The inverse query engine as recited in claim 32, wherein the cache is maintained within predefined limits by removing expired filters from a set of filters stored in the cache.

35. (Original) The inverse query engine as recited in claim 34, wherein an expired filter is a filter having an expiration time associated therewith that is earlier than a current time.

36. (Original) The inverse query engine as recited in claim 34, wherein an expired filter is a filter that has been stored in the cache for at least a specified period of time.

37. (Original) The inverse query engine as recited in claim 32, wherein the cache is maintained within predefined limits by removing at least one filter from a group of filters stored in the cache that has been used less recently than other filters in the group of filters.

38. (Original) The inverse query engine as recited in claim 37, wherein the at least one filter is removed only if the filter does not have an indication associated

therewith that identifies the filter as a permanent filter that is not to be removed from the cache.